University of Pennsylvania

ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

January 1995

The Problem of Signal and Symbol Integration: A Study of Cooperative Mobile Autonomous Agent Behaviors

Ruzena Bajcsy
University of Pennsylvania

Jana Košecká
University of Pennsylvania

The Problem of Signal and Symbol Integration: A Study of Cooperative Mobile Autonomous Agent Behaviors

Abstract

This paper explores and reasons about the interplay between symbolic and continuous representations. We first provide some historical perspective on signal and symbol integration as viewed by the Artificial Intelligence (AI), Robotics and Computer Vision communities. The domain of autonomous robotic agents residing in dynamically changing environments anchors well different aspects of this integration and allows us to look at the problem in its entirety. Models of reasoning, sensing and control actions of such agents determine three different dimensions for discretization of the agent-world behavioral state space. The design and modeling of robotic agents, where these three aspects have to be closely tied together, provide a good experimental platform for addressing the signal-to-symbol transformation problem. We present some experimental results from the domain of cooperating mobile agents involved in tasks of navigation and manipulation.
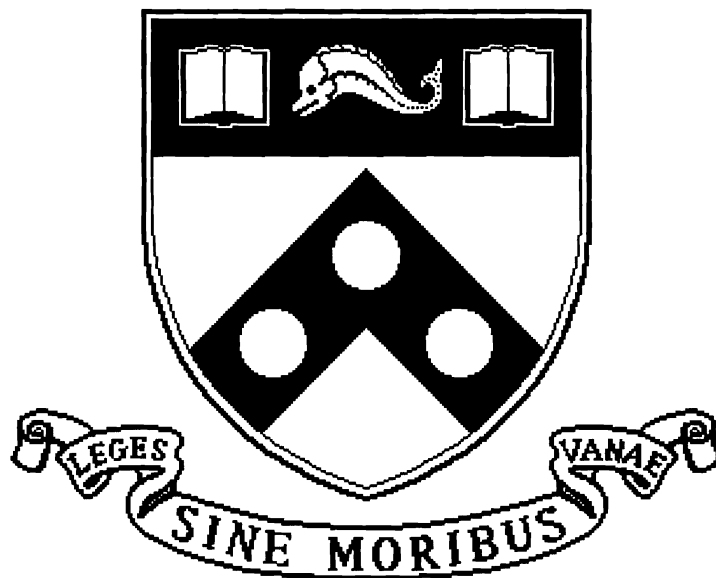
Comments

The Problem of Signal and Symbol Integration:
A Study of Cooperative Mobile Autonomous Agent Behaviors

MS-CIS-95-26
GRASP Lab 395

Ruzena Bajcsy and Jana Košecká

1995

The Problem of Signal and Symbol Integration: A Study of Cooperative Mobile Autonomous Agent Behaviors

Ruzena Bajcsy and Jana Košecká

GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania
3401 Walnut Street, 301C
Philadelphia, PA 19104

Abstract. This paper explores and reasons about the interplay between symbolic and continuous representations. We first provide some historical perspective on the signal and symbol integration as viewed by the Artificial Intelligence (AI), Robotics and Computer Vision communities. The domain of autonomous robotic agents residing in the dynamically changing environments anchors well different aspects of this integration and allows us to look at the problem in its entirety. Models of reasoning, sensing and control actions of such agents determine three different dimensions for discretization of the agent-world behavioral state space. The design and modeling of robotic agents, where these three aspects have to be closely tied together, provide a good experimental platform for addressing the signal-to-symbol-to-signal transformation problem. We present some experimental results from the domain of cooperating mobile agents involved in tasks of navigation and manipulation.

1   Introduction

To motivate the main topic of this paper, we begin with the assumption that agents live, behave and carry out certain tasks in a physical and dynamically changing environment. The issue we want to reason about here is one of representation and modeling of autonomous agents. We wish to argue that one needs a mixture or hybrid representation of signals and symbols. The question that remains, however, is what constitutes the *right* mixture.

Signal and symbol integration and transformation is an old but difficult problem. It comes about because the world surrounding us is a mixture of continuous space time functions with discontinuities. Recognition of these discontinuities in the world leads to representations of different states of the world, which in turn place demands on the agents behavioral strategies. Similarly, agent's (biological or artificial) closed loop interactions with the world/environment can be modeled as a continuous process, where as switching between different behaviors is naturally discrete. Furthermore, the tasks that are either externally given to the agents or internally self-imposed prespecify and, hence, discretize an other-

wise continuous behavior. Thus, we have three sources for discretization of the agent-world behavioral space:

1. Natural space-time discontinuities of the world.
2. The model of agent-world dynamics during execution of a given task.
3. The task.

Different subdisciplines dealing with the design and modeling of intelligent autonomous systems have addressed the problem described above differently. In the past, most Computer Vision focused on *signal-to-symbol* transformation, often called "pixels-to-predicates" as summarized in [Pen86]. The approach was to partition the signal into something "meaningful," in the geometric and photometric sense. Thus edges, lines, corners, regions of different shapes, and eventually three-dimensional objects and their shapes were recovered. Symbols served mainly as a data reduction mechanism.

From the early days of Artificial Intelligence, the importance of symbolic representations was continually emphasized by the founders of AI [Min63, McC68, NSS63] and their disciples. Unfortunately, the following was missing from this line of research:

1. Explicit acknowledgment that the transformation from signal to symbols results in the loss of information.
2. Self-correction and updating mechanisms of the obtained symbolic information.
3. Explicit models of the dynamic interaction between the agent and its world.

Concurrent with efforts at *signal-to-symbol* transformation, a *symbol-to-signal* endeavor was progressing in the Computer Vision community in the context of so-called top-down, knowledge-driven analysis of visual scenes [Win70]. The symbol/label represented the object of inquiry. This representation implied specific "procedures" that should be applied to the data in order to extract the expected features determined by a particular domain of visual scenes. *A priori* known symbolic information guided the selection of lower level matching methods. Detected symbolic primitives were then further used for reasoning about the spatial relationships among them in order to infer some higher level symbolic information. The commitment to the use of such *a priori* information eliminated the possibility to go back to the signal for resegmentation, or relabeling, depending on the higher-level reasoning triggered by some contradictions or inconsistencies.

Another instance where *symbol-to-signal* transformation occurs is related to the task-specific aspects of an agent. Various task specification languages were proposed, where, in the case of robotic applications, the symbolic representation of actions and goals was typically translated into continuous sensorimotor processes. These in turn specified particular control strategies for the available actuators and data acquisition and processing strategies for sensors [LP82, Lyo93, Bro93]. The aspects of programming and specifying the agents' tasks

were extensively investigated in more general task domains in AI, but the symbols rarely carried the information needed for detailed motion planning of the robot [Nil92, Fir92, Sch92, Kae90, Ark87].

In the mid 80's another development in the *signal-to-symbol-to-signal* debate was originated by Brooks [Bro86, Bro92] who, motivated by insect behaviors, challenged the prevalent view in AI of the necessity of symbolic representation as a precondition for "intelligent" behavior. He and his followers argued for the behavioral approach as a basis for constructing more complex autonomous "intelligent" behaviors. This brings again to the forefront the following questions:

> *Do intelligent systems need symbols? If so, what are they? How many symbols do they need? Are symbols innate or learned? Are they just ad hoc definitions or can they be derived in some systematic way, depending in task, context and environment?*

In this paper, we shall outline our recent ideas and understanding of this problem of *signal-to-symbol-to-signal* transformation in the context of the design and control of autonomous intelligent agents involved in cooperation. We shall begin with the definition of what we mean by *symbol* and what it implies for the design of autonomous systems. We shall then present the currently available mathematical models, which can guide the selection of symbols and finally provide some examples from the domain of cooperative mobile agents engaged in navigation and manipulation.

2 Problem Definition

2.1 What is a symbol?

Since a symbol will be some abstraction of a signal, let us, just for the purpose of setting the notation, refresh the standard notation used in the theory of dynamical systems (without committing ourselves to any particular system at this time). The state equations of a general continuous time-invariant dynamic system can be written as follows:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{1}$$

$$\dot{\mathbf{z}}(t) = g(\mathbf{x}(t), \mathbf{u}(t)) \tag{2}$$

where the first set of equations corresponds to a set of state equations, with initial conditions specified, $\mathbf{x}(t)$ is the time-varying state vector, and $\mathbf{u}(t)$ is the input control vector. The second set of equations corresponds to a set of output or measurement equations where $\mathbf{z}(t)$ is an output or measurement vector. Linear time-invariant systems with the following form are most frequently encountered and analyzed:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{3}$$

$$\dot{\mathbf{z}}(t) = C\mathbf{x}(t) \tag{4}$$

where A, B, C are real constant matrices. Also commonly encountered are affine nonlinear systems characterized as:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{u}(t)) \tag{5}$$

$$\dot{\mathbf{z}}(t) = h(\mathbf{x}(t)) \tag{6}$$

In the case of autonomous agents one is interested not only in modeling their behavior but also controlling them in order to achieve desired objectives. From this perspective [Bro88] one can divide control actions into two categories, namely *open-loop* control:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{u}(t))$$

where control vector $\mathbf{u}$ is constant over some period of time ignoring the measurements $\mathbf{z}$ and *closed-loop* control:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{u}(t) + k(h(\mathbf{x}(t))))$$

where the control becomes a function k of the observed measurements in the current state of the system. The application of different control laws would then correspond to the achievement of different objectives of the system, which in turn would be related to the task.

Our definition of a symbol has two different flavors. The first is a *descriptive* one, where a symbol represents a particular measurement vector $\mathbf{z}$. The second is a *procedural* one, denoting a set of strategies for extracting the measurements h, open-loop control strategies g or feedback control laws k to be applied. Within this setting the measurement vector, measurement function, and both open-loop and closed-loop control laws are dependent on the task, while the function f is related to the current model of the system. This may also change, however, in case the task constrains the number of degrees of freedom available. The measurement strategies together with the control strategies form the behaviors. The data/measurements come from either the environment via the perceptual apparatus of the agent or from its memory. The control and measurement strategies are encoded in terms of commands-symbols which invoke particular perception and action processes.

2.2 Why do we need symbols?

The need for symbols is partially motivated by the definition of the symbol in the previous section, where a *symbol* provides an abstraction of the workings of the low-level data acquisition and control strategies. The additional need and benefit of introducing a symbol is the "meaningful" task-related reduction and categorization of the sensory data, which can be further used for:

1. Abstraction and generalization.
2. Communication.
3. Memory-storage.
4. Reasoning.

The signal-to-symbol problem. We view this transformation as finding an equivalence class. In other words, it is a mapping of signal values into a sets of symbols. This implies data reduction (which is desirable) but also loss of information (which is undesirable). Hence, the question is a matter of determining the optimal granularity, or the number of descriptors/symbols, necessary for maximum data reduction and minimum loss of information.

The symbol-to-signal problem. As indicated above there are two cases: when the symbol is a command to invoke a behavior and when the symbol represents a measurement (parameter) vector, which can be supplied to a particular strategy. The transformation of symbol to signal is encoded in the semantics of the symbol, which is intimately related to the signal-to-symbol transformation capabilities of an agent.

2.3 How many symbols?

Given the need for symbols, the next question is "How many symbols are needed?" We approach this problem less philosophically from the point of view of robotics and autonomous systems. Our agents are characterized by the number of degrees of freedom (represented by the generalized coordinates) they possess given their sensory, mobility and manipulation capabilities. They live and interact with a physical environment obeying the laws of Newton's mechanics. The geometric and physical characteristics (i.e., both kinematic and dynamic models) of an agent are modeled only once. However, depending on the tasks and the types of constraints provided by the environment they are subject to change. This process of imposing constraints on the dynamics of the agent-environment interaction generates the first discretization of the behavioral space which otherwise can be considered a continuous space of general motion of the agent-environment system. This results in employing different degrees of freedom for the given task/strategy. For example the constraints imposed by the task and the environment in case of manipulation depend on the geometric properties of the objects being manipulated, therefore changing the number of degrees of freedom of the system (e.g., inserting a pin through two planes sliding with respect to each other reduces the number of degrees of freedom). Even more so for certain manipulation tasks, where the number of kinematic linkages can change (e.g., inserting a pin into a hole creates a new linkage with one rotational and one translational degree of freedom).

We postulate that for robotic agents the dynamic models of all their degrees of freedom are *a priori* given with a procedural capability to impose constraints based on either sensory information during interaction with the environment or coming from the task. Hence, symbols depend on the task.

2.4 Symbols and the task

We will elaborate on these issues in greater detail, centering our discussion about selection of symbols around different tasks. We shall consider two different cat-

egories of tasks, navigation and manipulation and provide a more detailed description of signal-symbol-signal transformation process in section 4.

Navigation tasks Navigation tasks involve perception of free space, places, objects, other agents and their spatial relationships. The task of navigation typically consists of two stages, first finding a path to the desired location and then finding a control law which would follow the path. If the potential field based approach [Kha86] to navigation is used, the stage of finding a path and following it can be merged into one stage and the desired control law computed as a gradient of a given potential function. Hence the task of the agent can be described by two discrete symbols, one representing the particular potential function representing the environment and the other representing the desired destination. This assertion holds in the case of static environments, where the global information about the environment is, *a priori*, available. In the case when the global information is not available and the robotic agent has to rely on local sensing capabilities, we consider the potential function with some generic form which is conveniently parameterized by the sensed local properties of the environment. Some ideas along these lines have been proposed by [Kod95].

This again brings up the question of how many symbols are sufficient. The answer entirely depends on the complexity of the environment in which the agent resides. There are two different types of information which need to be extracted from the environment for successful navigation: goals or landmarks to be detected and obstacles to be avoided. Goals and landmarks play a dual role. In the case of perfect position information, the goals can be simply specified in some global coordinate system. However, in order to achieve reliable position information, landmarks (or other *a priori* known features) are often used for localization. If the task is given externally to the agent or is self-imposed, the granularity of the prescribed path and thereby the richness of the symbolic vocabulary depends on the complexity of the environment.

Manipulation tasks Manipulation tasks, as in navigation, involve perception of free space, places, objects, other agents and their spatial relationship. However, the details about the space, place and objects obtained from perception need to be much finer than for navigation tasks. For example, for a grasping task, the size of the object and identification of graspable places is important. While during the task of manipulating an object one can divide the behavioral space into three steps (approach or move to the object, grasp, and manipulate), this division clearly does not imply only three symbols for control. The reason is that the approach and grasping very much depend on the specific manipulation subtask, which in turn depends on the geometric properties of the object. For example, the approach and grasp will be different if the manipulation task is only to lift the object and transport it to another place than if the task is to mate the object with another object [Lev95]. In other words, the ultimate purpose or function of the task and the complexity of the environment dictates the number of different control and sensing strategies, i.e., symbols. Finally, it should be self-evident that

the interplay between non-contact observations and contact perception during the execution of these tasks is much tighter than during navigation.

2.5 Distributed tasks

For general tasks the level and the type of symbolic information needed becomes more explicit when the tasks involve cooperation and coexistence of multiple agents. Both navigation and manipulation tasks may be simple instances of such tasks. For example when the agents have to march together while keeping in certain formation or have to grasp and carry a large object, while navigating in cluttered environments. In order to accomplish these cooperative tasks the agents need to share common goals and have the capabilities of either sensing the necessary information or communicating to each other beliefs about the state of the task/environment. Within this general setting one can trivially state that cooperation implies communication. Communication between two agents can, more specifically, take place through:

1. The environment.
2. Contact sensors (being in touch).
3. Non-contact observations (e.g. visual, ultrasound, infrared sensing).
4. A communication channel.

In the first three modes of cooperation, the agents do not need additional symbols since they are already part of their individual control and sensing strategies. In the fourth case however an additional symbol expressing an action of communication (sending and receiving a message) needs to be established. Distributed cooperative tasks bring out various interesting issues regarding tradeoffs between communication (in the sense of establishing a communication channel) and sensing. A more formal treatment of this subject can be found in [DJR93, BS95].

3 Mathematical Models

The choice of descriptive symbols determining the state of the system and the model of the interaction with the environment is determined by the physical characteristics of the system/agent. Modeling of these aspect has been studied extensively in the theory of dynamic systems and control theory. In the following paragraph we will give a brief overview of such modeling principles.

We assume that our agents can be modeled as multiple degree of freedom (DOF) mechanistic systems that interact with the physical world that obeys Newton's law of physics. We also assume that our agents are equipped with contact force sensors, position sensors and non-contact vision and ultrasound sensors.

3.1 Dynamic model of the agent

Under these assumptions, we follow the Lagrangian formulation of dynamics [BH95, Cra89] for multiple degree of freedom systems. The formulation of equations of motions is built around the basic principle of virtual work, which states that the work done by all forces is equal to zero:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i \tag{7}$$

where F_i is an external force corresponding to the generalized coordinate q_i. Lagrangian $L(q, \dot{q})$ in the previous formulation is:

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) \tag{8}$$

where $T(q, \dot{q})$ is the overall kinetic energy of the system and $U(q)$ is potential energy, both expressed in generalized coordinates system $(q_0, q_1, ..., q_n)$, where n is the number of degrees of freedom of the system. The dynamic equations are then:

$$F(q) = M(q)\ddot{q} + C(q, \dot{q}) + G(q) \tag{9}$$

where $M(q)$ is inertia matrix, $C(q, \dot{q})$ is the matrix of Coriolis and centrifugal effects, vector $G(q)$ denotes gravity terms and F is the generalized force vector. These equations determine what work the agent must exert in order to carry out a motion under the conditions determined by the inertia of the agent body, Coriolis forces and gravitational forces. In case some external constraints imposed by interaction with the world/environment and/or by the task are present, they are captured by the Lagrangian coefficients λ weighted by the matrix $A(q)$. The equations then become:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = F(q) + A^T(q)\lambda. \tag{10}$$

Similarly, the constraints of various types of interaction can be incorporated into this equation. In the case of a more complicated system, such as mobile manipulators, first the dynamic equation for the manipulators and mobile platform are established individually and then the mutual effects of manipulator and mobile platform are added as extra terms into the equations (e.g., inertia terms caused by platform rotation are added as additional forces to the dynamic equation of the manipulator) [Yam94].

The Lagrangian framework provides a powerful modeling tool for mechanical systems where the geometric and physical properties of the system are well understood and easily describable. However, difficulties arise once again when it comes to modeling the constraints provided by the environment, especially when they have to be extracted by sensors. These issues have been extensively studied in the area of manipulation, where the geometric and physical properties of objects to be manipulated provide additional constraints [Mas82, LP82]. The models as described so far do not explicitly use the information extracted from sensory data, in a different form other than the information of a current state of the system q, which is assumed to be available at each instant of time. An

example of a different type of modeling, where the sensed information about the dynamically changing environment is directly part of dynamic equations is outlined in the following paragraph.

3.2 Dynamic model of perception-action cycle

The idea of incorporating the model of the environment into the dynamic model of the agent is very appealing and has been extensively addressed by several neuroscientists looking at problems related to motor control [Sch91]. The general ideas regarding models of action-perception patterns come from a series of experiments for control of posture in the presence and absence of visual stimuli, time-to-contact and various tracking, grasping and catching behaviors.

In his extensive studies of dynamic action-perception patterns [Sch91], Schöner looked at the problem of control of posture, and demonstrated that the visual information stabilizes posture in the visual world. Schöner proposed a model of coupled oscillators, where the agent's intrinsic dynamics is modeled by second order linear system with an eigenfrequency ω_0 and the visual appearance of the environment is modeled by a environment function $e(x, t)$. The behavior of the postural control system can be described in simplest mathematical form as:

$$\ddot{x} + \alpha \dot{x} + (\omega_0)^2 - \sqrt{Q}\xi_t = -c_{env}e(x, t) \tag{11}$$

where ξ_t is Gaussian white noise, Q is the strength of the noise and $e(x, t)$ represents the expansion rate of the target in retinal coordinates. For sinusoidally moving surround $e(x, t)$ the solution to the postural response is a harmonic with the same frequency as the visual motion. The system can be studied by transforming Equation (11) into polar coordinates and looking at the relative phase of the two components. These two systems are naturally coupled and the system can be described in terms of relative phase Φ dynamics by the following equation:

$$\dot{\Phi} = A + B\sin(\Phi) + \sqrt{Q}\xi \tag{12}$$

where A and B are constants representing, relating the eigenfrequency ω_0, driving frequency of the stimulus ω_d. For more details see [DSGG94]. As shown, this equation is nonlinear and the coefficients A and B are measures of how much the two oscillators are phase locked, corresponding to how much the agent's behavior is in harmony with the visual stimulus that reflects the environment. It should be obvious that the nature of Equation (12) will be different depending on the environmental function $e(x, t)$, which can again give rise to the variety of symbols. In the previous example, the function $e(x, t)$ was a periodic function expressed in the coordinate system of the observer. Not only is there more information contained in the optical flow [Koe86] which could be subjected to similar analysis, but one can also employ different sensing modalities for investigating stability properties of action-perception couplings.

In a slightly different setting, formal modeling of action-perception systems has been extensively studied in the visual servoing literature, see [Has93] for an overview.

4 Task Description Language

Various mathematical models outlined in the previous section provided us with some insights into the problem of what a symbol is an abstraction of. We recognized two inherently different categories of symbols: one representing the state vector of the mechanical system or the environment and the other one representing the procedural aspect of the interaction of the agent with the environment. This determines the set of symbols which are necessary for a given physical agent, residing in particular environment, engaged in particular tasks.

A set of signals/symbols defines all the capabilities of the agent. These comprise a set of elementary control strategies for available actuators and a set of perceptual strategies. Determining the set of elementary control strategies is determined by the agents "physique," while the set of perceptual strategies is more task dependent. While carrying out the tasks, there is typically a large number of processes/strategies activated in parallel, interacting with each other and the environment. It is very important to be able to understand and characterize these interactions in a general fashion in order to develop modular and easily extendable systems which can be employed for a variety of tasks.

In order to facilitate the symbol to signal transformation, as well as propose some design guidelines for characterizing robot behaviors, which depend on the task, we propose a language for specifying tasks, where tasks are characterized as networks of processes. This representation was originally proposed in [Lyo93]. However, instead of adopting the semantics of basic schemas in terms of port automata, we propose to model the elementary strategies in terms of Finite State Machines (FSM's). This representation is very intuitive and straightforward, providing a clear abstraction for a variety of already existing control and perceptual strategies. Moreover, the representation is further amenable to formal analysis. We are able to synthesize a discrete event controller, which serves as monitor and run-time scheduler for the task. For details of this procedure see [Koš95].

For modeling purposes each elementary strategy or computation is represented as a process[1] and has a FSM model associated with it. The transitions between the states of the FSM model are modeled by events, clearly capturing initiation, termination, interruption or change of global variables (settings) of the elementary strategy. The global variables (or more specifically predicates on them) play a general role in our framework, expressing the goals the robot should achieve, maintain or prevent from happening. The set of final states of elementary strategies is partitioned into a set of successful and unsuccessful final states. Communication between two processes running in parallel is modeled via shared events. If the two processes share an event then a communication link between them is established.

Elementary processes are combined together by a set of composition operators. The operators (common to almost any process model) capture the temporal and structural dependencies between the processes. As we mentioned earlier,

---

[1] The word *process* and *strategy* will be used interchangeably.

since the types of behaviors which need to be invoked depend on the task to be accomplished, we adopt the notion of the task representation as a network of processes. Processes can be composed in a sequential fashion (R ; S), where S starts after R terminates, in a concurrent fashion ($R \parallel S$), where R and S run in parallel, in a conditional fashion ($R < v >$: $S(v)$), where S starts after R terminates successfully computing v, which is then used to initialize process S, and in a disabling fashion ($R \sharp S$), which is similar to parallel composition except that if one of the processes terminates the other process is terminated as well. Two additional composition operators expressing repetitive behavior are synchronous recurrent composition (R :; S), defined recursively as R :; $S = R : (S ; (R :; S))$, and asynchronous recurrent composition ($R < v > :: S$), defined recursively as R :; $S = R : (S \parallel (R :: S))$.

We shall demonstrate some of these ideas in the tasks of both individual agent navigation and cooperative multiple agent navigation in the presence of obstacles.

4.1 Applications and results

The task of navigation for one or two mobile bases requires a basic control strategy for achieving a goal in an environment cluttered with obstacles, sensing capabilities for obstacle detection, communication capabilities between the two robots, and a strategy for achieving an arbitrary heading. For the time being we assume that the desired goal location is given in a global coordinate system relative to the starting point of the robot and that the reading from the position encoders corresponds to a correct position of the robot in the world coordinate system. The navigational capabilities are implemented using potential field based control, where the control law for reaching a desired location is derived as a gradient of a given potential function with minimum at the goal configuration. This formulation follows nicely from Lagrangian formulation (equation (9)) where control of the mechanical system is based on the selection of the force F as a command vector. This is the basic idea behind the potential field methods for task planning and control pioneered by [Kha86]. In order to achieve the desired control objective in a given environment the force has to capture the aspects of the environment necessary for achieving the control objective. One way to look at this problem is from the point of view of optimization [Kod92], where one represents the environment in terms of some cost function which attains a value at each point in the configuration space and has a global minimum at the desired target location. The vector field of this cost function then corresponds to the force field in which the mechanical system resides. Applying at each point the particular force vector then leads the system to the desired location. For the task of navigation in the examples presented here we will not use the full dynamical model of the mobile base, but assume a simple kinematic model of a omnidirectional mobile robot. The configuration of the base is denoted by $X = (x, y)$ and the goal location $X_g = (x_g, y_g)$ is represented by an attractive potential field:

$$U_a(X) = \frac{k_p}{2}(X_g - X)^2$$

where k_p is a constant gain factor. In order to achieve the desired goal we need to exert a force, which is proportional to the negative gradient of the given potential function $F = -\nabla(U_a(X))$. The encountered obstacles are represented by a hyperbolic repulsive potential function:

$$U_r(X) = \begin{cases} \frac{k_r}{\gamma}(\frac{1}{\eta(X)} - \frac{1}{\eta_o})^\gamma & \text{if } \eta(X) \le \eta_0 \\ 0, & \text{otherwise} \end{cases}$$

where coefficient $\gamma > 2$, η is a distance function to the obstacle, η_0 is the obstacle's influence range and k_γ is a constant gain factor. Instead of using the traditional gradient field we adopt the vortex field [MO91] of the above function representing the field rotating around the obstacle:

$$F_v(X) = \pm \begin{bmatrix} \frac{\partial U_r(X)}{\partial y} \\ -\frac{\partial U_r(X)}{\partial x} \end{bmatrix}.$$

The desired velocity $\dot{X}_d = (\dot{x}_d, \dot{y}_d)$ at each instance of time is derived from the artificial potential field holonomic path planner as:

$$\dot{X}_d = -\nabla(U_a(X) + U_r(X)) \tag{13}$$

or alternatively using the vortex field method as:

$$\dot{X}_d = -\nabla U_a(X) + F_v(X). \tag{14}$$

This basic control strategy is parameterized by the environment, where the desired goal location determines the shape of potential function U_a and the obstacles detected along the way are captured by the function U_r. Parameters of these functions are provided by perceptual processes. This control strategy will later be referred to by the symbol GoTo. Once initiated, the control law is applied until the desired location is reached. The strategy can fail in a number of predictable ways, such as when the goal is in the location of the obstacle or some mechanical failure of the robot occurs.

The sensing strategy for detecting obstacles in our system is based on inverse perspective mapping [MBLB91]. The obstacles are approximated by an ellipse and the correspondences between the obstacles are established in consecutive frames [Koš95]. This perceptual routine provides the necessary parameters for the GoTo strategy, by updating the potential function U_r. We refer to this perceptual strategy, in the next example, by the symbol Detect.

The control strategy for two mobile robots marching in formation, denoted March, is similar to the GoTo strategy, with an additional parameter of the distance between the two robots and a rule for generating the commands to the individual bases. The elementary strategy for reaching the desired heading is denoted by GoToHeading.

In this example the task is for two mobile bases A and B to go individually to a predetermined location while avoiding obstacles, wait for each other, align, and then go to another predetermined location, marching in a side-by-side formation, while avoiding obstacles. This task can be expressed in a task specification language in the following manner:

$$((GoTo_A(Goal_1) \, \natural \, Detect_A) \quad \natural \quad (GoTo_B(Goal_2) \, \natural \, Detect_B)) :$$
$$(Align_A(Heading) \quad \natural \quad Align_B(Heading)) :$$
$$((March_A(Goal_3) \, \natural \, Detect_A) \quad \natural \quad (March_B(Goal_3) \, \natural \, Detect_B))$$

Disabling composition is used because the overall task requires cooperation so in the case one of the basic strategies fails the others terminate as well. If two strategies are invoked in a parallel or disabling manner and they share certain events, a communication link between them is established. This not only allows the Detect strategy to update the parameters of the GoTo strategy, but also allows the two agents to share the information about the obstacles between them while they are marching in side-by-side formation (see Figure 1).
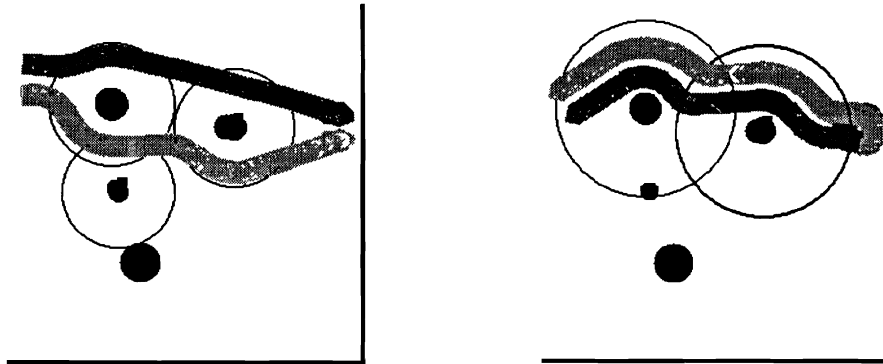

Fig. 1. The agents are first told to individually go to a predetermined location while avoiding obstacles and wait for each other (left). They then march together in a side-by-side formation toward a desired goal (right). Notice that while they are marching together they "agree" to avoid the obstacle from the same side in spite of the fact that it would be more advantageous for one of the agents to navigate to the left of the obstacle. The sensitivity region around the obstacle, which triggers the avoidance maneuver is proportional to the distance between the agents.

Another example demonstrates a more global navigation task, where the agent is told to go to a desired location via route specified in terms of places which need to be visited along the way. Navigation between two consecutive places is achieved by visual servoing on that particular place. The vision component of the strategy denoted by symbol Track provides the desired heading direction to the potential field navigation function GoTo. In order to initialize the tracking

routine, the target is first localized via perceptual recognition strategy **Look**. The overall task of passing through the door, heading towards down the hallway and the heading towards another door (see Figure 2) can be specified as follows:

$$((Look(Door_1) : Track(Door_1)) \natural (GoTo_A())) :$$
$$((Look(Passage_1) : Track(Passage_1)) \natural (GoTo_A())) :$$
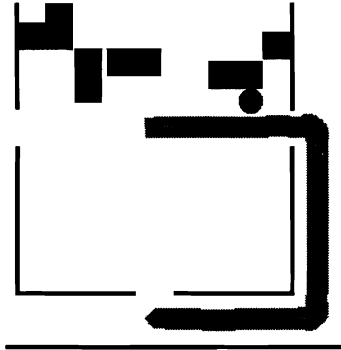$$((Look(Door_2) : Track(Door_2)) \natural (GoTo_A()))$$

Fig. 2. The task of the agent is to go through the doorway on the right, down the corridor and continue to the doorway on the left. Navigation between consecutive places done by visual servoing on the particular features which characterize the place.

The examples outlined above demonstrate that, for navigational tasks, an agent uses a set of elementary control and perceptual strategies. These strategies together with their parameters constitute a set of necessary symbols. The strategies can then be composed based on the task to be accomplished and parameterized based on the environment. The task specification language formalizes this composition and guarantees run-time scheduling and monitoring of the task.

5 Conclusion

Symbols do not come for granted, but their meaning is deeply embedded in modeling the low-level interactions of the agent with the environment. A thorough understanding of these aspects provides us with insights into the low-level workings of the system, understand failures and guarantee success, all in the presence of uncertain and noisy information. For elementary tasks, which can be achieved by a unique composition of perceptual and control strategies, agents can be modeled and described purely in terms of differential equations. However, the need for symbols is inevitable if one wants to build and model agents involved in a

variety of tasks and environments. Symbols not only provide nice abstractions for low-level strategies, but also allow us to move one level up the modeling hierarchy and observe the properties of the systems and their interactions between each other and their environment at a more macroscopic level. Symbolic representation mediates reasoning about the sequential and repetitive nature of various tasks and allows specification of interactions and communications between multiple agents in distributed systems. The need for symbols is not the only message we are trying to deliver here. Studying systems which sense and interact with real-world environments and engage in a variety of tasks provides an excellent platform for understanding the fine line between the task-related aspects of modeling and the innate elementary sensory and motion capabilities of the agents.

References

[Ark87]   R. C. Arkin. Motor schema based navigation for a mobile robot. In *IEEE Proc. Intl. Conf. on Robotics and Automation*, 1987.

[BH95]   A. E. Bryson and Y. C. Ho. *Applied Optimal Control.* Hemisphere Publishing Coorporation, 1995.

[Bro86]   R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14 – 23, March 1986.

[Bro88]   R. W. Brockett. On the control of movement. In *IEEE Proceedings on Robotics and Automation*, pages 534–540, 1988.

[Bro92]   R. A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. *A.I. Memo 1091*, February 1992.

[Bro93]   R. W. Brockett. Hybrid models for motion control systems. In *Perspectives in Control*, pages 29–54. Birkhauser, 1993.

[BS95]   R. I. Brafman and Y. Shoham. Knowledge considerations in robotics and distribution of robotics tasks. In *to appear in Proceedings IJCAI*, 1995.

[Cra89]   John J. Craig. *Introduction to Robotics: Mechanics and control.* Addison-Wesley, 1989.

[DJR93]   B. Donald, J. Jennings, and D. Rus. Experimental information invariants for cooperating autonomous mobile robots. In *Workshop on Dynamically Intercating Robots, IJCAI*, 1993.

[DSGG94] T. M. Dijkstra, G. Schöner, M. A. Giese, and C. M. Gielen. Frequency dependence of the action-perception cycle for postural control in a moving visual environment: relative phase dynamics. *Biological Cybernetics*, (71):489–501, 1994.

[Fir92]   J. R. Firby. Building symbolic primitives with continuous control routines. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, pages 62 – 69, 1992.

[Has93]   K. Hashimoto. *Visual Servoing.* World Scientific, 1993.

[Kae90]   L. P. Kaelbling. An architecture for intelligent reactive systems. In A. Tate J. Allen, J. Hendler, editor, *Readings in Planning*, pages 713–729. Morgan Kaufman Publishers, 1990.

[Kha86]   O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[Kod92]   D. Koditschek.   Robot planning and control via potential functions. *Robotics Review*, 1992.

[Kod95]   D. Koditschek. The geometry of a robot programming language. In *First Workshop on Algorithmic Foundations of Robotics*, 1995.

[Koe86]   Jan J. Koenderink. Optic flow. *Vision Research*, 26(1):161–179, 1986.

[Koš95]   J. Košecká. Supervisory control of autonomous mobile agents. Technical report, GRASP Laboratory, Departments of Computer Science, University of Pennsylvania, 1995.

[Lev95]   E. Levison. Object specific reasoning. Technical report, Dissertation Proposal, Department of Computer Science, University of Pennsylvania, 1995.

[LP82]   T. Lozano-Perez.   Task planning.   In M. Brady, J. H. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, editors, *Robot Motion: Planning and Control*, pages 463–489. MIT Press, 1982.

[Lyo93]   D. M. Lyons. Representing and analyzing action plans as networks of concurrent processes. *IEEE Transactions on Robotics and Automation*, 1993.

[Mas82]   M. Mason. Compliance and force control for computer controlled manipulators. In M. Brady, J. H. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, editors, *Robot Motion: Planning and Control*. MIT Press, 1982.

[MBLB91] H.A. Mallot, H.H. Bulthoff, J.J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185, 1991.

[McC68]   J. McCarthy. Semantic information processing. In M. Minsky, editor, *Programs with Common Sense*, pages 403–410. MIT press, 1968.

[Min63]   M. Minsky. Steps toward artificial intelligence. In E. feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 406–450. McGraw-Hill Book Company, New York, 1963.

[MO91]   C. De Medio and G. Oriolo. Robot obstacle avoidance using vortex fields. In S. Stifter and J.Lenarcic, editors, *Advances in Robot Kinematics*, 1991.

[Nil92]   N. J. Nilsson. Toward agent programs with circuit semantics. Technical report, Department of Computer Science, Stanford University, 1992.

[NSS63]   A. Newell, J. Shaw, and H. Simon. Empirical explorations of the logic theory machine. In E. feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 406–450. McGraw-Hill Book Company, New York, 1963.

[Pen86]   A. Pentland. *From Pixels To Predicates*. Ablex Publishing Company, Norwood, NJ, 1986.

[Sch91]   G. Schöner. Dynamic theory of action-perception patterns: the moving room paradigm. *Biological cybernetics*, (64), 1991.

[Sch92]   M. J. Schoppers. How to interface high level planner with low level continuous control. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, 1992.

[Win70]   P. H. Winston. *Learning Structural Descriptions from examples*. PhD thesis, MIT, 1970.

[Yam94]   Y. Yamamoto. *Control and Coordination of Locomotion and Manipulation of a Wheeled Mobile Manipulator*. PhD thesis, University of Pennsylvania, Grasp Laboratory, 1994.